MATLAB: poznámky ...

Olga Majlingová

# Obsah

1	MA	TLAB
	1.1	Instalace (na ČVUT)
	1.2	
		1.2.1 Hlavní Menu
		1.2.2 Okna MATLABu
		1.2.3 Získání nápovědy
	1.3	
		1.3.1 Proměnné a konstanty
		Typ a hodnota proměnné
		Smazání proměnné
		Uložení proměnných do souboru
		Nahrání proměnné (z .mat-souboru)
		Import dat x Excelu

# Cvičení 1

# MATLAB

MATLAB umožňuje snadnou a rychlou práci s maticemi reálných nebo komplexních čísel. Zároveň umí vypočítané výsledky graficky znázornit. Zpracování výsledků nám usnadní také mnoho hotových funkcí, stačí si jen pročíst (anglickou) nápovědu. Na rozdíl od používání Excelu a jiných aplikací MS Office, kde většinu činností provádíme klikáním myší, v MTALABu většinu příkazů píšeme na klávesnici.

## 1.1 Instalace (na ČVUT)

Po přihlášení na

https://download.cvut.cz

najdete soubory ke stažení, také návod a fórum.

https://download.cvut.cz/tag/technical-programs/

 Licenční server je přístupný pouze ze sítě ČVUT nebo po připojení přes VPN.
 Připojení z domova (VPN)

https://download.cvut.cz/cisco-anyconnect-4-5







Na fakultních počítačích spouštíme Matlab kliknutím na ikonku

### 1.2 Popis prostředí

Po spuštění systému se objeví okno složené z několika částí (viz obr.1.1).



Obrázek 1.1: okna MATLABu

#### 1.2.1 Hlavní Menu

Z Menu v horním řádku zmíníme pouze několik pojmů záložky HOME.

	HOME	PLOTS	APPS	EDITOR	PUBLISH V	/IEW					6	-12 📫	~ C 🖸 ? -	Q Search Documentation
	New New Script Live Scrip	New Ope	Find Files	Import Save Data Workspan	New Variable	Favorito	Analyze Code	Simulink	Layout	③ Preferences Set Path Parallel ▼	Add-Ons	? Help	23 Community 3 Request Support ■ Learn MATLAB	
1		FILE			VARIABLE		CODE	SIMULINK		ENVIRONMENT			RESOURCES	

New Script, New, Open vytvoří (otevře) nový soubor pro editování skriptu.



použijeme pro importování dat do Matlabu (například z Excelu).

Uspořádání oken můžeme změnit, například v menu Layout

Velikost písma nastavíme (změníme) v menu Preferences OPreferences, kde vybereme Fonts, viz obrázek 1.2.

MATLAS	MATLAB Fonts Preferences Desktop code font Currently used by: Command Window, Command History, Editor					
App Designer Code Analyzer						
<ul> <li>Colors</li> <li>Command History</li> <li>Command Window</li> <li>Comparison</li> <li>Current Folder</li> <li>Editor/Debugger</li> </ul>	Monospaced	Plain 2 13				
	Sample The quick brown fox jumps over the lazy dog. 1234567890					
Tab Language Code Folding Backup Files	Desktop text font Currently used by: Current Folder, Wo Use system font	rkspace, Variables, Function Browser				
Autoformatting	Lucida Grande	0 Plain 0 13				
Autocoding	Sample					
# Fonts	The quick brown fox jumps over the lazy dog. 1234567890					
Custom > General GLIDE Help > Keyboard Toolbars						
	Help browser font					
	To change the font size for the Help and Web browsers, right-click on the page and choose Zoom In or Zoom Out. This setting will be saved across sessions.					
Variables	Custom fonts					
Workspace	Currently used by: Profiler					
Simulink	To set a custom font for any desktop	tool, go to the Custom Fonts preferences.				
Computer Vision System Toolbox						
Image Acquisition Toolhox						
Computer Vision System Toolbox Database Toolbox Image Acquisition Toolbox		Of Cracel And				

Obrázek 1.2: Nastavení písma



Clear Commands umožňuje vymazání obsahu oken Command Window a Command History.

#### 1.2.2 Okna MATLABu

**Okno Current Folder.** V tomto okně se zobrazuje obsah aktuálního (pracovního) adresáře. Aktuální adresář změníme šipkami nebo zadáním cesty.  $\Leftrightarrow \textcircledightarrow \textcircledightarrow \textcircledightarrow \verbNATLAB</code>, Ke změně lze také využít kliknutí na <math>\textcircledightarrow \textcircledightarrow menu.$ 

Okno Workspace. Zde se zobrazuje informace o všech dostupných proměnných pracovního prostředí.

#### Okno Command History. V tomto okně se zobrazují všechny použité příkazy.

Každé spuštění MATLABu je označeno datem a časem (zelený text mezi procenty). Příkazy z tohoto okna lze znovu spustit dvojklikem na vybraný příkaz, případně upravit přetažením příkazu myší do Command Window. Historii lze vymazat kliknutím na šipku v pravém rohu okna (kontextové menu) a výběrem Clear Command History. Okno Command History nemusíme používat, protože v Command Window lze listovat použitými příkazy s použitím šipek ( $\uparrow$ ,  $\downarrow$ ).

**Okno Command Window.** V tomto okně pracujeme v režimu dialogu a do tohoto okna se zobrazuje průběh výpočtu, některé výsledky nebo hlášení o chybě.

ans =

5

Za přihlašovací >> napíšeme příkaz a odešleme klávesou enter ( $\overleftrightarrow$ ). Pokud je příkaz zapsán správně, je proveden, případně se zobrazí výsledek.

>> 2 + 3

>> 2 + 4;

Pokud je v zápisu příkazu chyba, zobrazí se hlášení.

>> ě + 2

Pokud není uvedeno, kam se má uložit výsledek, uloží se do proměnné ans.

Po provedení uvedených příkazů se v okně **Workspace** objevila proměnná ans, v okně **Command History** se zaznamenaly příkazy.

Workspace		$\overline{\mathbf{v}}$	Command History	$\overline{\mathbf{v}}$
Name A	Value 6		% 20/09/2019, 11:12% 2 + 3 2 + 4; - ě + 2	

Je-li příkaz ukončen středníkem, výsledek se nezobrazí.

Po provedeném >> 2 + 3 se výsledek uložil do proměnné ans a zobrazil se v Command Window. Po >> 2 + 4 se výsledek uložil do proměnné ans, její hodnota se změnila, ale v Command Window se nezobrazil. V zápisu >> ě + 2 se objevila chyba, příkaz se neprovedl, hodnota proměnné ans se nezměnila.

**Zobrazení čísel.** Čísla se implicitně zobrazují se čtyřmi znaky za desetinnou tečkou. Způsob **zobrazování výsledků** si lze zvolit pomocí příkazu **format**, například:

>> format compact kolem výsledků nebudou prázdné řádky,
>> format long čísla se nebudou vypisovat se čtyřmi, ale se čtrnácti desetinnými místy,
>> format rat čísla se budou zobrazovat ve tvaru zlomku.
K původnímu nastavení (zobrazení čtyřmi se znaky za desetinnou tečkou) se vrátíme příkazem >> format nebo

>> format short.

Obsah tohoto okna lze vymazat popsanou volbou Clear Commands z menu nebo příkazem >> clc.

**Okno Editor** slouží k editování programu (skriptu), uloženého v textovém souboru. V textovém souboru zapisujeme příkazy. Editované příkazy se neprovádějí, začnou se provádět po spuštění skriptu. V záložce menu **EDITOR** lze vytvořit nový skript, otevřít existující, uložit změněný. Pomocí menu Comment lze několik příkazů zakomentovat (označit jako komentář) nebo odkomentovat. Na levé straně okna editoru se zobrazuje čtvereček červené , oranžové , nebo zelené barvy. Je-li tam zelený čtvereček, je možné skript spustit. Je-li červený,

musíme nejprve opravit syntax v označných řádcích. Skript lze spustit z okna editoru kliknutím na ikonku

#### 1.2.3 Získání nápovědy

**Funkce help** vyhledá informaci o hledané funkci, pokud taková existuje, zobrazí tuto informaci do Command Window. Například:

hledaná funkce je nalezena

```
>>help inv
inv Matrix inverse.
    inv(X) is the inverse of the square matrix X.
    A warning message is printed if X is badly scaled or
    nearly singular.
    See also slash, pinv, cond, condest, lsqnonneg, lscov.
    Reference page for inv
    Other functions named inv
>>
• hledaná funkce není nalezena
```

>> help kuku

kuku not found.

Use the Help browser search field to search the documentation, or type "help help" for help command options, such as help for methods.

Funkce lookfor vyhledá hledaný pojem ve všech souborech nápovědy.

>> lookfor XYZ

vyhledá řetězec XYZ v prvním řádku komentáře ve všech dostupných matlabovských souborech.

>> lookfor XYZ -all

vyhledá řetězec XYZ v prvním bloku komentáře ve všech dostupných matlabovských souborech.

**Funkce doc** otevře **Okno Nápovědy**, hypertextově uspořádané webové stránky, příklad je obrázku 1.3. K nápovědě v tomto tvaru se dostaneme i zapsáním hledaného výrazu ve vyhledávacím okně menu **Případně můžeme kliklnout na odkaz Reference** page for ..., který se objeví po výpisu výsledku volání help.

•••	Help	
Other uses of help		×
Documentation	Search Help	Q
< Documentation Home	help	
< MATLAB	Help for functions in Command Window	collapse all in page
< Desktop Environment		
< Help and Support	Syntax	
< MATLAB < Functions	help help name	
help	Description	
ON THIS PAGE	help lists all primary help topics in the Command Window. Each main	n help topic

Obrázek 1.3: Okno nápovědy

## 1.3 Základní prvky jazyka

#### Data a příkazy

- DATA: (vstupní hodnoty, mezivýsledky, výsledky) proměnné a konstanty
- PŘÍKAZY: co umí vykonavatel algoritmu, jak mění hodnoty proměnných
- POŘADÍ PŘÍKAZŮ: za sebou, skoky (větvení, opakování)

Změnu hodnot proměnných provádíme přiřazovacím příkazem.

Podmíněný příkaz, přepínač a příkazy cyklu určují pořadí příkazů.

#### 1.3.1 Proměnné a konstanty

Proměnné mají jméno, typ a hodnotu.

Jméno (název, identifikátor) je posloupnost písmen a číslic, která začíná písmenem.

Správná např

Za písmena považujeme a..z A..Z (bez diakritiky), \_ (znak podtržítka). NIC JINÉHO ! Číslice jsou 0..9. Malá a velká písmena se rozlišují, to znamená, že prom, PROM, Prom jsou 3 různé proměnné. Maximální počet znaků proměnné je určen funkcí namelengthmax, lze zjistit například

pocet\_znaku = namelengthmax

Chybná jména proměnných jsou například

J	J	J = = =	Spruvnu nupr.
• 10	CV	identifikátor nesmí začínat číslicí,	<ul> <li>cv1</li> </ul>
■ C\	/-1	pomlčka nepatří mezi písmena,	■ c1v
■ C\		tečka nepatří mezi písmena,	<ul> <li>cv_1</li> </ul>
■ C\	/1	mezera nepatří mezi písmena.	<ul> <li>Cv1</li> </ul>

Jméno proměnné nesmí být shodné s žádným z klíčových slov, jako jsou například while, if, end a další. Seznam klíčových slov lze zjistit pomocí funkce iskeyword. Použijeme-li pro proměnnou jméno již existující funkce, mohou při výpočtu nastat velmi podivné věci. V Matlabu existují jména i, j pro práci s komplexními čísly, pi pro přibližnou hodnotu čísla  $\pi$ , eps a další. Pokud si nejsme jisti, lze pomocí funkce exist ověřit, zda jméno již je používáno. Pokud se stane, že předefinujeme existující jméno funkce, smažeme námi definovanou proměnnlou clear jmeno a původní význam se "vrátí".

#### Typ a hodnota proměnné

**Symbolické proměnné.** Matlab má specifický typ proměnné – symbolické. Proměnné tohoto typu musíme před jejich použitím deklarovat syms x y . Ostatní proměnné nedeklarujeme, vznikají při prvním zadání hodnoty. Obecně jsou všechny proměnné považovány za **matice** složené z prvků stejného typu.

Typ proměnné je určen hodnotou, která je v proměnné uložena.

Prvky matic mohou být různých typů:

<ul> <li></li></ul>	🛛 🤳 struktura
<ul> <li>Iogické konstanty</li> </ul>	🔹 🔝 buňka (cell)
<ul> <li>Image: state of the st</li></ul>	■ 🗐 datum ■ 🖽 tabulka

Různým typům hodnot odpovídají různé ikonky, které se zobrazují ve Workspace.

Typ hodnoty určuje množinu operací (funkcí), které na proměnné lze aplikovat.

#### Vytvoření proměnné

Proměnná vznikne zadáním hodnoty. Zadat hodnoty lze: • přířazovacím příkazem:

•	přiřazovacím příkazem:		$n\acute{azev\_prom\check{e}nn\acute{e}} = v\acute{yraz}$	1	a :	= 1
•	z klávesnice pomocí funkce	input()	a	a =	input(' nějaký text	')
•	ze souboru pomocí funkce	load()	a	a =	<pre>load('jmeno_souboru</pre>	')
	Data uložená v souborech .	xlsx lze do Matlab	u <b>importovat</b> .			

<sup>&</sup>lt;sup>1</sup>Zatím se omezíme na nejjednodušší výraz, kterým je konstanta. Více o výrazech bude uvedeno dále.

Čísla. Desetinná čísla zadáváme s desetinnou tečkou (ne čárkou!) nebo pomocí zlomku, například -8.14 1/2. Pokud je před desetinnou tečkou jenom nula, lze ji vynechat, například .02 místo 0.02. Je možné používat vědecký formát, stejně jako v Excelu, například 1.6e11 znamená  $1.6 \cdot 10^{11}$ , 6.122e-8 znamená  $6.122 \cdot 10^{-8}$ . Komplexní čísla. Imaginární část zadáváme s použitím i anebo j (i, j jsou předdefinovaná jména), například komplexní číslo 2 + 4i lze zadat jako 2+4i, -0.32 - 0.000015i zapíšeme -.32-1.5e-5i. Předdefinovanými jmény jsou také:

- realmax  $\doteq 1.7977 \cdot 10^{308}$ .
- realmin  $\doteq 2.2251 \cdot 10^{-308}$ ,
- eps  $\doteq 2.2204 \cdot 10^{-16}$ ,
- **pi** = 3.141592653589793,
- Inf reprezentuje +∞ a vzniká, když výsledkem operace je "příliš velké" číslo, např.1/0, ale i 1.e1000,
- NaN (Not-A-Number) je výsledkem nedefinované matematické operace, např  $\infty \infty$ ,  $0 \cdot \infty$ ,  $\frac{0}{0}, \frac{\infty}{\infty}$ .

Logické konstanty jsou true a false. Zobrazují se také jako logical 1 a logical 0.

**Znakové konstanty** se zapisují v jednoduchých apostrofech, např. 'a'.

**Řetězcové konstanty** se zapisují v dvojitých apostrofech, např. "ahoj". Rozdíl mezi 'ahoj' a "ahoj"je ten, že 'ahoj' je vektor 4 znaků, zatímco "ahoj"je jeden řetězec, tj. vektor rozměru  $1 \times 1$ .

**Příklady** vytvoření jednoduché proměnné (vektor rozměru  $1 \times 1$ )

a = 3	e = true	
b = -1.2345	f = false	k = 'a'
c = 15e-2	g = logical(1)	l = "ahoj"
d = 2-5i	h = logical(0)	

**Vytvoření vektoru nebo matice.** Matice  $m \times n$  vytváříme pomocí hranatých závorek, do kterých zapisujeme prvky matice po řádcích. Jednotlivé řádky oddělujeme středníkem, prvky v řádku oddělujeme mezerou nebo čárkou. Zadání matice po prvcích:

- řádky (vektor 1 × 4) a = [1 2 3 4] nebo b = [1, 2, 3, 4]
- sloupce (vektor 4 × 1) c = [1 2 3 4]' nebo d = [1; 2; 3; 4]
- matice  $(4 \times 2)$  A = [1 2; 3 4; 5 6; 7 8]

K jednotlivým prvkům matice přistupujeme pomocí jména proměnné a indexů. Pro vektory používáme jeden index, pro matici dva. Prvek v i-tém řádku, j-tém sloupci zapisujeme A(i,j), A(4,2) = 15 změní hodnotu prvku matice A ve 4.řádku a 2. sloupci.

Pokus manipulovat s prvkem mimo matici t = A(4,5) (zprava od rovnítka) vyvolá chybové hlášení: Index in position 2 exceeds array bounds (must not exceed 2), ale A(4,5) = 9 (zleva od rovnítka) změní rozměr matice, tato matice bude mít o 2 sloupce více, dosud nezadané prvky budou mít hodnotu 0.

2	0	0	0
4	0	0	0
6	0	0	0
15	0	0	9
	2 4 6 15	2 0 4 0 6 0 15 0	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

**Struktura** slouží k uložení do jedné skalární proměnné (jednoho prvku matice) několika druhů údajů, které spolu souvisí. Například bod může mít souřadnice, student – jméno, ročník, předměty, známky. Jednotlivé položky lze pojmenovat a přistupovat k nim pomocí tečky. Příklad struktury student:

```
student.jmeno = ['Jan', 'Jakub']
student.rocnik = 2
student.predmety = ["ZAPG", "M3"]
student.znamky = [1 2]
```

Všimněte si rozdílu mezi vektorem znaků jmeno a vektorem řetězců predmety: student.jmeno je vektor z 8 znaků, student.jmeno(1) je 'J',..., student.jmeno(8) je 'b', zatímco student.predmety je vektor ze 2 řetězcu, student.predmety(1) je "ZAPG", student.predmety(2) je "M3".

**Buňky** používáme, když chceme do jedné matice uložit hodnoty různých typů. Buňky zapisujeme ve složených závorkách, například C = {'ahoj', [56 67 78]} vytvoří vektor C složený ze 2 buněk, jehož první prvek je buňka obsahující vektor znaků, druhý prvek je buňka obsahující vektor čísel.

- C(1) je 1 x 1 cell array {'ahoj'}
- C{1}(1) je 'a', C{1}(2) je 'h', C{1}(3) je 'o', C{1}(4) je 'j'
- C(2) je 1 x 1 cell array {1 x 3 double}
- C{2}(1) je 56, C{2}(2) je 67, C{2}(3) je 78
- chybné pořadí závorek: C(2){1}, C(1){2}

Datum umožňuje lepší práci s časovými údaji.

Tabulka umožňuje seskupit více údajů.

#### Zobrazení hodnoty proměnné

Chceme-li zjistit obsah proměnné (tj. její hodnotu), můžeme použít

středník Některé příkazy vypisují své výsledky (např. x = 215).

Pokud nepotřebujeme tyto výsledky vidět, lze potlačit výpis výsledků příkazu napsáním středníku na konec příkazu

(x = 215;).

Vypisování výsledků výpočtu by mělo být přehlednější než to, co vznikne pouhým nenapsáním středníku.

disp Nejméně užitečná je funkce disp, která se chová přibližně stejně, jako vynechání středníku.

fprintf Elegantní úpravu vytvoříme pomocí funkce fprintf
 (více - help fprintf).

save Obsah jednotlivých proměnných lze uložit do souboru pomocí funkce save (více - help save)

printf Obrázky (grafy) lze uložit v různých grafických formátech pomocí funkce print (více - help print).

publish Skript (text programu) můžeme uložit v různých formátech pomocí funkce publish (více - help publish).

#### Smazání proměnné

Proměnné, které už nechceme používat, lze smazat pomocí příkazu clear . clear název\_proměnné - smaže vybranou proměnnú2 název\_proměnnéN - smaže vybrané proměnné (jejich názvy od sebe oddělujeme mezerou) clear - smaže všechny proměnné z Workspace. Příklad: clear b a (smaže proměnnou b a proměnnou a) clear (smaže všechny proměnné)

Neuvedeme-li seznam proměnných, smažou se všechny. Smazané proměnné nelze obnovit.

#### Uložení proměnných do souboru

Někdy se stane, že potřebujeme uchovat nějakou důležitou proměnnou pro použití při příštím spuštění MATLABu (např. musíme přerušit práci). Potřebujeme tedy proměnnou dostat z paměti na disk počítače. K tomu slouží příkaz save :

save jméno\_souboru název\_proměnné - do zadaného souboru se uloží vybraná proměnná,

save jméno\_souboru název\_proměnné1 název\_proměnné2 název\_proměnnéN - do zadaného souboru se uloží N vybraných proměnných (jejich názvy od sebe oddělujeme mezerou),

save jméno\_souboru - do zadaného souboru se uloží všechny proměnné z Workspace.

Ve všech případech je vytvořen soubor s příponou .mat (ta je přidána automaticky), který obsahuje názvy a hodnoty vybraných proměnných.

Příklad: uložení dvou proměnných a a maticeA do souboru 'pokus.mat':

save pokus a maticeA (soubor 'pokus.mat' vznikne v pracovním adresáři)

Příkaz save má širší možnosti - viz help save.

#### Nahrání proměnné (z .mat-souboru)

Je-li na disku nějaký soubor s uloženou jednou nebo více proměnnými (vznikl pomocí save), lze tyto proměnné dostat do pracovního prostředí MATLABu pomocí příkazu load.

Příklad: clear load pokus

Příkaz load má širší možnosti - viz help load.

#### Import dat x Excelu



Import se provede kliknutím na Import Data. Importovaná data se uloží do matice nebo vektoru v aktuálním pracovním kontextu (workspace).

Je možné zadat rozsah hodnot např. Range A2:Qxxx.

Je vhodné použít Numeric Matrix nebo Column vectors.



-11-

## Seznam použité literatury

- [1] PECINOVSKÝ, J., PECINOVSKÝ, R. Office 2016 průvodce uživatele, Grada, 2016, 256 s. ISBN 978-80-9052-2(pdf).
- [2] PECINOVSKÝ, J. Microsoft Office Podrobná uživatelská příručka, Computer Press, 2015, 496 s. ISBN 978-80-251-4300-1.
- [matlab] DOŇAR, B., ZAPLATÍLEK, K. MATLAB pro začátečníky, 1.díl, BEN Technická literatura, 2. vydání, 2003, 152 s.
- [Atwood (2009)] ATWOOD, J. HTML Validation: Does It Matter?, Coding Horror, 2009, [online] [cit.2017-07-18] Dostupné z: https://blog.codinghorror.com/html-validation-does-it-matter/